Digital Signatures

Murat Kantarcioglu

Digital Signatures

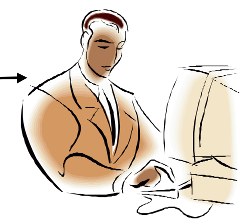
- Define a digital signature scheme $DS = (\mathcal{K}, Sign, VF)$
- Key generation: $(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}$
- Signing a message: $\sigma \stackrel{\$}{\leftarrow} Sign_{sk}(M)$
- Signature Verification $d \stackrel{\$}{\leftarrow} VF_{pk}(M, \sigma)$
  - $d = 1$ if σ is valid for for given message under (pk, sk) pair
  - else $d = 0$

Digital Signature Assumptions

Alice generates *(pk,sk)*

Bob has correct pk

$$(M, \sigma \leftarrow Sig_{sk}(M))$$



Bob outputs $VF_{pk}(M, \sigma)$

- Bob assumed to have correct pk
- Sender (Alice) has the private key
- Sig could be randomized and /or stateful
- We will mainly focus on deterministic Sig algorithms
  - Unlike PKE algorithms

Defining Security

Definition 9.2 Let $\mathcal{DS} = (\mathcal{K}, \text{Sign}, \text{VF})$ be a digital signature scheme, and let A be an algorithm that has access to an oracle and returns a pair of strings. We consider the following experiment:

Experiment $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{DS}}(A)$
$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}$
$(M, \sigma) \leftarrow A^{\text{Sign}_{sk}(\cdot)}(pk)$
If the following are true return 1 else return 0:
– $\text{VF}_{pk}(M, \sigma) = 1$
– $M \in \text{Messages}(pk)$
– M was not a query of A to its oracle

The *uf-cma-advantage* of A is defined as

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS}}(A) = \Pr\left[\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{DS}}(A) = 1\right] . \blacksquare$$

PK

(M, σ) for any M

A:
M_1, σ_1
M_2, σ_2
.
M_q, σ_q

$M \in \{M_1 \ldots, M_q\}$

UTD

RSA based Signatures

- $((N,e),(N,p,q,d)) \leftarrow (K)$ where $e.d = 1 \bmod \phi(N)$, $N = pq$

- Signature Generation
  - Algorithm $Sign_{N,p,q,d}(M)$
  - if $M \in Z_N^*$ return $\perp$
  - return $M^d \bmod N$

- Verification
  - Algorithm $VF_{N,e}(M,\sigma)$
  - if $M \cancel{\in} Z_N^* \vee \sigma \cancel{\in} Z_N^*$ return 0
  - if $M = \sigma^e \bmod N$ return 1 else 0

- Direct RSA signature generation is not secure

Possible Attacks

$Adv^{uf\text{-}cma}_{DS}(F_1) = 1$

- ★ Forger F_1
  - ▶ Forger $F_1^{Sign_{N,p,q,d}()}(N, e)$
  - ▶ return $(1, 1)$

$1^d \bmod N = 1$

$(1, 1)$

All attacks have advantage one

- ★ Forger F_2
  - ▶ Forger $F_2^{Sign_{N,p,q,d}()}(N, e)$
  - ▶ $\sigma \leftarrow Z_N^*$, $M \leftarrow \sigma^e \bmod N$
  - ▶ return (M, σ)

$Adv^{uf\text{-}cma}_{DS}(F_2) = 1$

M^d, σ

$\Rightarrow (\sigma^e)^d = \sigma$

- ★ Forger F_3
  - ▶ Forger $F_3^{Sign_{N,p,q,d}()}(N, e)$
  - ▶ $M_1 \leftarrow Z_N^* - \{1, M\}$, $M_2 \leftarrow M M_1^{-1} \bmod N$
  - ▶ $\sigma_1 \leftarrow Sign_{N,p,q,d}(M_1), \sigma_2 \leftarrow Sign_{N,p,q,d}(M_2)$
  - ▶ return $(M, \sigma_1\sigma_2 \bmod N)$

Hash-then-invert paradigm

- Goal: RSA based scheme that
  - is provably secure
  - has Flexible message space
- Idea Hash the message first given $H_N : \{0,1\}^* \mapsto Z_N^*$
- Signature Generation
  - Algorithm $Sign_{N,p,q,d}(M)$
  - $y \leftarrow H_N(M)$
  - return $y^d \bmod N$
- Verification
  - Algorithm $VF_{N,e}(M, \sigma)$
  - $y \leftarrow H_N(M)$
  - if $y = \sigma^e \bmod N$ return 1 else 0

Hash then Invert Paradigm

- Previous Forgers described do not work well for Hash-then-Invert
  - $H_N(1) \neq 1$ with high probability (w.h.p)
  - $\sigma^e \bmod N \neq H_N(M)$ w.h.p
  - $H_N(M_1).H_N(M_2) \neq H_N(M)$ w.h.p
- Not secure if it is easy to find $M_1 \neq M$ such that $H_N(M_1) = H_N(M_2)$
- What are the assumptions needed to make Hash then Invert Paradigm Secure??

Full Domain Hash RSA signatures

- $H : \{0,1\}^* \mapsto Z_N^*$ is a random function known by everybody

- Signature Generation
  - Algorithm $Sign_{N,p,q,d}^{H(.)}(M)$
  - $y \leftarrow H(M)$
  - return $y^d \bmod N$

- Verification
  - Algorithm $VF_{N,e}^{H(.)}(M, \sigma)$
  - $y \leftarrow H_N(M)$
  - if $y = \sigma^e \bmod N$ return 1 else 0

Experiment $\mathbf{Exp}_{\mathcal{DS}}^{\text{uf-cma}}(F)$

$((N,e),(N,p,q,d)) \stackrel{\$}{\leftarrow} \mathcal{K}_{\text{rsa}}$

$H \stackrel{\$}{\leftarrow} \mathsf{Func}(\{0,1\}^*, \mathbf{Z}_N^*)$

$(M,x) \stackrel{\$}{\leftarrow} F^{H(\cdot),\mathsf{Sign}_{N,p,q,d}^{H(\cdot)}(\cdot)}(N,e)$

If the following are true return 1 else return 0:

- $\mathrm{VF}_{\mathrm{pk}}^{H}(M,\sigma) = 1$
- M was not a query of A to its oracle

FDH-RSA

- Consider adversaries running in time t, making q_{sig} oracle queries and at most q_h hash queries

- Simulate the random H by choosing random answers and storing them on a table
  - Function $H(x)$
  - If $T(x) \neq$ Null Then $T(x) \xleftarrow{\$} Z_N^*$
  - Return $T[x]$

- Thm: Let FDH-RSA in the random oracle model described as befor
  Let F be an adversary attacking FDH-RSA making q_{siq} signature queries, q_h hash queries. Then $\exists$ an Adversary I

$$Adv_{DS}^{uf-cma}(F) \leq q_h . Adv_{K_{rsa}}^{ow-kea}(I)$$

Proof of Thm

- Note I is given $(N, e), y$ and tries to find x s.t. $x^e \bmod N$

  $y = x^e \mod N$

- I will $\boxed{\text{run } F}$ to find the $\underline{x}$

- I will answer Fs oracle queries to H and $Sign$ as it wishes

- I will use the F to invert y

- Idea: I modifies answers to Fs oracle queries to invert y

Proof of Thm

Inverter $I(N, e, y)$

  Initialize arrays $Msg[1 \ldots q_{\text{hash}}]$, $X[1 \ldots q_{\text{hash}}]$, $Y[1 \ldots q_{\text{hash}}]$ to empty

  $j \leftarrow 0$; $i \xleftarrow{\$} \{1, \ldots, q_{\text{hash}}\}$

  Run F on input (N, e)

  If F makes oracle query (hash, M)

    then $h \leftarrow \mathit{H\text{-}Sim}(M)$; return h to F as the answer

  If F makes oracle query (sign, M)

    then $x \leftarrow \mathit{Sign\text{-}Sim}(M)$; return x to F as the answer

  Until F halts with output (M, x)

  $y' \leftarrow \mathit{H\text{-}Sim}(M)$

  Return x

y'

$H\text{-}Sim(M)$

$\underline{H(M),\ x^d \bmod N}$

$Msg[j]$	–	The j-th hash query in the experiment
$Y[j]$	–	The reply of the hash oracle simulator to the above, meaning the value playing the role of $H(Msg[j])$. For $j = i$ it is y.
$X[j]$	–	For $j \neq i$, the response to sign query $Msg[j]$, meaning it satisfies $(X[j])^e \equiv Y[j] \pmod N$. For $j = i$ it is undefined.

Proof of Thm

M

Subroutine $H\text{-}Sim(v)$
  $l \leftarrow Find(Msg, v, j)$; $j \leftarrow j+1$; $Msg[j] \leftarrow v$
  If $l = 0$ then
    If $j = i$ then $Y[j] \leftarrow y$
    Else $X[j] \xleftarrow{\$} Z_N^*$; $Y[j] \leftarrow (X[j])^e \bmod N$
    EndIf
    Return $Y[j]$
  Else
    If $j = i$ then abort
    Else $X[j] \leftarrow X[l]$; $Y[j] \leftarrow Y[l]$; Return $Y[j]$
    EndIf
  EndIf

$X[j] \leftarrow r$ $Y[j] \leftarrow r^e \bmod N$

Subroutine $Sign\text{-}Sim(M)$
  $h \leftarrow H\text{-}Sim(M)$
  If $j = i$ then abort
  Else return $X[j]$
  EndIf

★ $Find(A, v, j)$
- if $\nexists l \leq j, A[l] = v$ return 0
- else smallest l where $A[l] = v$

$A = [\overset{l'}{1}, 3, 3, 5, 7]$

$Find(A, 3, 1) = 0$

Find (A, v, j)
{
  For (i=1 to j)
    if (A[i] == v)
      return i;
  return 0;
}

$Find(A, 3, 3) = 2$

Proof of Thm.

- Inside $H - sim(v)$, if $l = 0$ and $j \neq i$ $X[j] \leftarrow Z_N^*$ and $Y[j] \leftarrow (X[j])^e \bmod N$ and returns $Y[j]$
- Sign-sim(M) returns $X[j]$

$y \leftarrow H(M) \xrightarrow{} d$
$y = x^d \bmod n$

$$
\begin{aligned}
Pr[I \text{ inverts } y] &= Pr[I \text{ inverts } y | \text{ no abort }].Pr[\text{ no abort }] \\
&\quad + Pr[I \text{ inverts } y | \text{ abort }].Pr[\text{ abort }] \\
&= Pr[I \text{ inverts } y | \text{ no abort }].Pr[\text{ no abort }] \\
&\geq Adv_{DS}^{uf-cma}(F).\frac{1}{q_{hash}}
\end{aligned}
$$

when $\underline{I}$ calls the H-sim last time

if Find (Msg, M, q_{hash}) = l & $l = i$)

PSS0

- $H : \{0,1\}^* \mapsto Z_N^*$ is a random function known by everybody

- Signature Generation
  - Algorithm $Sign_{N,p,q,d}^{H(\cdot)}(M)$
  - $r \xleftarrow{\$} \{0,1\}^s$
  - $y \leftarrow H(r||M)$
  - return $(r, y^d \bmod N)$

$H(M)$, $\boxed{H(r||M)}$

- Verification
  - Algorithm $VF_{N,e}^{H(\cdot)}(M, \sigma)$
  - Parse σ as (r, x)
  - $y \leftarrow H(r||M)$
  - if $y = x^e \bmod N$ return 1 else 0

Theorem 9.4 Let $\mathcal{DS}$ be the PSS0 scheme with security parameters k and s. Let F be an adversary making $q_{\rm sig}$ signing queries and $q_{\rm hash} \geq 1 + q_{\rm sig}$ hash oracle queries. Then there exists an adversary I such that

$$\mathbf{Adv}_{\mathcal{DS}}^{\text{uf-cma}}(F) \leq \mathbf{Adv}_{\mathcal{K}_{\rm rsa}}^{\text{ow-kea}}(I) + \frac{(q_{\rm hash}-1)\cdot q_{\rm sig}}{2^s}\,. \quad (9.3)$$

El-Gamal Signature Scheme

- Define a digital signature scheme $DS = (\mathcal{K}, Sign, VF)$
- Key generation: $((p, \alpha, y), (p, a)) \xleftarrow{\$} \mathcal{K}$ Where $\alpha^a = y \bmod p$ and α is a generator of Z_p^*
- Signing a message M — $H(M||r)$
  - Select $k \in Z_p^*$ with $gcd(k, p-1) = 1$
  - $r \leftarrow \alpha^k$, $s \leftarrow k^{-1}(H(M) - ar) \bmod (p-1)$
  - return (r, s)

  secret key
- Signature Verification for $(M, (r, s))$
  - $v_1 \leftarrow y^r r^s \bmod p$
  - $v_2 \leftarrow \alpha^{H(m)} \bmod p$
  - Accept if $v_1 = v_2$

$y^r = \alpha^{ar} \bmod p$

$r^s = \alpha^{k \cdot (k^{-1}(H(M) - ar))}$

$y^r r^s = \alpha^{H(m) - ar} \alpha^{ar} \bmod p$

$= \alpha^{H(M)}$

The Digital Signature Algorithm (DSA)

★ Key Generation:

- Select a prime $2^{159} < q < 2^{160}$
- Choose $0 \le t \le 8$ and a prime p where $2^{511+64t} < p < 2^{512+64t}$ and $q|(p-1)$
- Select a random $b \in Z_p^*$ s.t. $\alpha \leftarrow b^{(p-1)/q}$ mod p and $\alpha neq 1$ mod p
- Select a random integer a s.t $1 \le a \le q-1$ $\qquad \alpha \not\equiv 1 \bmod p$
- Compute $y \leftarrow \alpha^a$ mod p
- Public key is (p, q, α, y), private key is a

★ Signature Generation: for message M

- Select a random k s.t. $0 < k < q$
- $r \leftarrow (\alpha^k \bmod p) \bmod q$ $\qquad s \leftarrow k^{-1}(H(M)+ar)$
- $s \leftarrow k^{-1}(H(M) + ar) \bmod q$
- return (r, s)

DSA

- Verification for $(M, (r, s))$
  - Check that $0 < r < q$ and $0 < s < q$
  - $u_1 \leftarrow s^{-1}.H(M)$ and $u_2 = rs^{-1} \bmod q$
  - $v \leftarrow (\alpha^{u_1}.y^{u_2} \bmod p) \bmod q$
  - Accept iff $v = r$

$s = k^{-1}(H(m) + ar)$

$r = (\alpha^k \bmod p) \bmod q$

$v \leftarrow \alpha^{s^{-1}.H(m)} . \alpha^{ars^{-1}} = \alpha^{s^{-1}(H(m)+ar)}$

$= \alpha^{(k^{-1}.(H(m)+ar))^{-1}.(H(m)+ar)}$

$= \alpha^{k\,(H(m)+ar)^{-1}.(H(m)+ar)}$

$= \alpha^k$

Schnorr Scheme

★ Key generation is the same as DSA except no restriction on (p, q)

★ Signature generation for M

- Choose random secret k, $1 \le k \le q-1$
- $r \leftarrow \alpha^k \bmod p$, $e \leftarrow H(M||r)$, $s \leftarrow ae + k \bmod q$
- return (s, e)

★ Signature verification for $(M, (s, e))$

- $v \leftarrow \alpha^s y^{-e} \bmod p$ and $e = H(M||v)$
- Accept iff $e = e$